Statistical Machine Learning (BE4M33SSU)
Lecture 7: Generative learning, EM-Algorithm

Czech Technical University in Prague

- Generative vs. Discriminative Learning
- Maximum Likelihood Estimator, consistency
- Expectation Maximisation Algorithm

1. Generative vs. Discriminative Learning

Generative learning:

- Model the **joint** probability distribution $p_\theta(x, y)$ for features $x \in \mathcal{X}$ and hidden states $y \in \mathcal{Y}$ up to unknown parameter(s) $\theta \in \Theta$.
- Optimal prediction strategy (if true parameter θ_0 is known):

$$h(x) \in \arg\min_{y \in \mathcal{Y}} \sum_{y' \in \mathcal{Y}} p_{\theta_0}(y' \mid x) \ell(y', y)$$

- Learning: if $\theta_0 \in \Theta$ is not known, estimate it from training data $\mathcal{T}^m = \{(x^j, y^j) \in \mathcal{X} \times \mathcal{Y} \mid j = 1, \ldots, m\}$ e.g. by maximum likelihood estimator (MLE)

$$\theta^* \in \arg\max_{\theta \in \Theta} \sum_{i=1}^{m} \log p_\theta(x^j, y^j)$$

1. Generative vs. Discriminative Learning

Discriminative learning(1):

- Model only the **conditional** distributions $p_\theta(y \mid x)$, $\theta \in \Theta$.
- Optimal prediction strategy (if true parameter θ_0 is known): as above
- Learning: if $\theta_0 \in \Theta$ is not known, estimate it by maximising the conditional likelihood of the training data $\mathcal{T}^m$.

$$\theta^* \in \arg\max_{\theta \in \Theta} \sum_{i=1}^{m} \log p_\theta(y^j \mid x^j)$$

Discriminative learning(2):

- Model the class of prediction strategies $h \in \mathcal{H}$.
- Optimal prediction strategy (if $p(x, y)$ is known):

$$h_0(x) = \arg\min_{y \in \mathcal{Y}} \sum_{y' \in \mathcal{Y}} p(y' \mid x) \ell(y', y)$$

- Estimate the optimal strategy $h^* \in \mathcal{H}$ by minimising the empirical risk on the training data

$$h^* \in \arg\min_{h \in \mathcal{H}} \frac{1}{m} \sum_{j=1}^{m} \ell(y^j, h(x^j))$$

1. Generative vs. Discriminative Learning

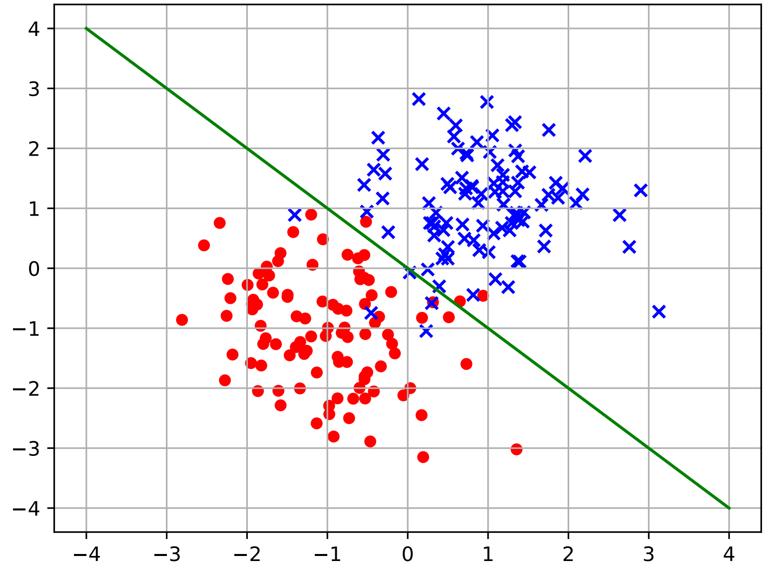
Example (Gaussian Discriminative Analysis, Logistic Regression, Linear Classifier)

$x \in \mathbb{R}^n$, $y \in \{0,1\}$ with $y \sim Bern(\alpha)$ and $x \mid y \sim \mathcal{N}(\mu_y, V)$, i.e.

$$p(y) = \alpha^y (1-\alpha)^{1-y}$$
$$p(x \mid y) = \frac{1}{(2\pi)^{n/2}|V|^{1/2}} \exp\Big[-\frac{1}{2}(x-\mu_y)\cdot V^{-1}\cdot(x-\mu_y)\Big]$$

Generative learning: Denote $I_1 = \{j \mid y^j = 1\}$ and I_0 correspondingly. ML estimator for training data $\mathcal{T}^m = \{(x^j, y^j) \mid j = 1, \ldots, m\}$ gives

$$\alpha^* = \frac{1}{m}|I_1|$$
$$\mu_0^* = \frac{1}{|I_0|}\sum_{j\in I_0} x^j, \quad \mu_1^* = \frac{1}{|I_1|}\sum_{j\in I_1} x^j$$
$$V^* = \frac{1}{m}\sum_{j=1}^{m}(x^j - \mu_{y^j}) \otimes (x^j - \mu_{y^j})$$



1. Generative vs. Discriminative Learning

Discriminative learning(1): Notice that the posterior conditional probabilities can be expressed as

$$p(y \mid x) = \frac{\exp[y(\langle w, x \rangle + b)]}{1 + \exp[\langle w, x \rangle + b]},$$

i.e. a logistic regression, where w and b are some functions of α, μ_0, μ_1 and V.

Estimate w and b by maximising the conditional likelihood on training data

$$(w^*, b^*) \in \arg\max_{w,b} \Big\{ \sum_{j \in I_1} (\langle w, x^j \rangle + b) - \sum_{i=1}^{m} \log\big(1 + \exp(\langle w, x^j \rangle + b)\big) \Big\}$$

The objective is concave in w and b. Its global optimum can be found by gradient ascent.

Discriminative learning(2): The optimal inference rule is a linear classifier. Learn it by minimising the empirical risk. $\Rightarrow$ SVM

1. Generative vs. Discriminative Learning

Question: The three methods will provide different decision boundaries when trained on the same dataset. Which one is better?

General answer:

- Generative learning makes stronger assumptions and is more data efficient when the assumptions are (nearly) correct.
- Discriminative learning makes weaker assumptions and is less data efficient but significantly more robust to deviations from model assumptions.

2. Consistency of the Maximum Likelihood estimator

Let $\mathcal{T}^m = \{z^j \mid j = 1,\ldots,m\}$ be i.i.d. generated from $p_{\theta_0}(z)$, with $\theta_0 \in \Theta$ unknown.

Which conditions ensure consistency of the MLE $\theta^* = \arg\max_{\theta\in\Theta} \log p_\theta(\mathcal{T}^m)$?

$$\mathbb{P}_{\theta_0}\big(\|\theta_0 - \theta^*(\mathcal{T}^m)\| > \epsilon\big) \xrightarrow{m\to\infty} 0$$

Denote log-likelihood of training data $L(\theta,\mathcal{T}^m) = \frac{1}{m}\sum_{i=1}^{m} \log p_\theta(z^j)$

and expected log-likelihood $L(\theta) = \mathbb{E}_{\theta_0}\big(L(\theta,\mathcal{T}^m)\big) = \sum_{z\in\mathcal{Z}} p_{\theta_0}(z) \log p_\theta(z)$

Consider $L(\theta,\mathcal{T}^m) = L(\theta) + \big[L(\theta,\mathcal{T}^m) - L(\theta)\big]$

- The model should be identifiable, i.e. $\theta_0 = \arg\max_{\theta\in\Theta} L(\theta)$
- Ensure that he Uniform Law of Large Numbers (ULLN) holds, i.e.

$$\mathbb{P}_{\theta_0}\big(\sup_{\theta\in\Theta} |L(\theta,\mathcal{T}^m) - L(\theta)| > \epsilon\big) \xrightarrow{m\to\infty} 0$$

for any $\epsilon > 0$.

2. Consistency of the Maximum Likelihood estimator

Identifiability of the model θ_0 is easy to prove if $p_{\theta_0}(z) \not\equiv p_\theta(z)$ holds $\forall \theta \neq \theta_0$.

Let $p(z), q(z)$ be two probability distributions s.t. $p \not\equiv q$. Then

$$\sum_{z \in \mathcal{Z}} p(z) \log p(z) > \sum_{z \in \mathcal{Z}} p(z) \log q(z)$$

follows from strict concavity of the function $\log(x)$:

$$-D_{KL}(p \parallel q) = \sum_{z \in \mathcal{Z}} p(z) \log \frac{q(z)}{p(z)} < \log \sum_{z \in \mathcal{Z}} \frac{q(z)p(z)}{p(z)} = \log 1 = 0$$

ULLN can be ensured e.g. by requiring that

- $L(\theta, z)$ is continuous in θ and $\Theta \subset \mathbb{R}^k$ is compact.
- $L(\theta, z)$ can be upper bounded: $\log p_\theta(z) \leqslant d(z)$ $\forall \theta$ with $\mathbb{E}_{\theta_0} d(z) < \infty$.

3. The Expectation Maximisation Algorithm

Unsupervised generative learning:

- The joint p.d. $p_\theta(x, y)$, $\theta \in \Theta$ is known up to the parameter $\theta \in \Theta$,
- given training data $\mathcal{T}^m = \{x^j \in \mathcal{X} \mid i = 1, 2, \ldots, m\}$ i.i.d. generated from p_{θ_0}.

How shall we implement the MLE

$$\theta^*(\mathcal{T}^m) = \arg\max_{\theta \in \Theta} \frac{1}{m} \sum_{x \in \mathcal{T}^m} \log p_\theta(x) = \arg\max_{\theta \in \Theta} \frac{1}{m} \sum_{x \in \mathcal{T}^m} \log \sum_{y \in \mathcal{Y}} p_\theta(x, y)$$

- If θ is a single parameter or a vector of homogeneous parameters $\Rightarrow$ maximise the log-likelihood directly.
- If θ is a collection of heterogeneous parameters $\Rightarrow$ apply the **Expectation Maximisation Algorithm** (Schlesinger, 1968, Sundberg, 1974, Dempster, Laird, and Rubin, 1977)

3. The Expectation Maximisation Algorithm

EM approach:

- Introduce auxiliary variables $\alpha_x(y) \geqslant 0$, for each $x \in \mathcal{T}^m$, s.t. $\sum_{y \in \mathcal{Y}} \alpha_x(y) = 1$
- Construct a lower bound of the log-likelihood $L(\theta, \mathcal{T}^m) \geqslant L_B(\theta, \alpha, \mathcal{T}^m)$
- Maximise this lower bound by block-wise coordinate ascent.

Construct the bound:

$$L(\theta, \mathcal{T}^m) = \frac{1}{m} \sum_{x \in \mathcal{T}^m} \log \sum_{y \in \mathcal{Y}} p_\theta(x, y) = \frac{1}{m} \sum_{x \in \mathcal{T}^m} \log \sum_{y \in \mathcal{Y}} \frac{\alpha_x(y)}{\alpha_x(y)} p_\theta(x, y) \geqslant$$

$$L_B(\theta, \alpha, \mathcal{T}^m) = \frac{1}{m} \sum_{x \in \mathcal{T}^m} \sum_{y \in \mathcal{Y}} \alpha_x(y) \log p_\theta(x, y) - \frac{1}{m} \sum_{x \in \mathcal{T}^m} \sum_{y \in \mathcal{Y}} \alpha_x(y) \log \alpha_x(y)$$

3. The Expectation Maximisation Algorithm

Maximise $L_B(\theta, \alpha, \mathcal{T}^m)$ by block-coordinate ascent:

Start with some $\theta^{(0)}$ and iterate

E-step Fix the current $\theta^{(t)}$, maximise $L_B(\theta^{(t)}, \alpha, \mathcal{T}^m)$ w.r.t. α-s. This gives

$$\alpha_x^{(t)}(y) = p_{\theta^{(t)}}(y \mid x).$$

M-step Fix the current $\alpha^{(t)}$ and maximise $L_B(\theta, \alpha^{(t)}, \mathcal{T}^m)$ w.r.t. θ.

$$\theta^{(t+1)} = \arg\max_{\theta \in \Theta} \frac{1}{m} \sum_{x \in \mathcal{T}^m} \sum_{y \in \mathcal{Y}} \alpha_x^{(t)}(y) \log p_\theta(x, y)$$

This is equivalent to solving the MLE for annotated training data.

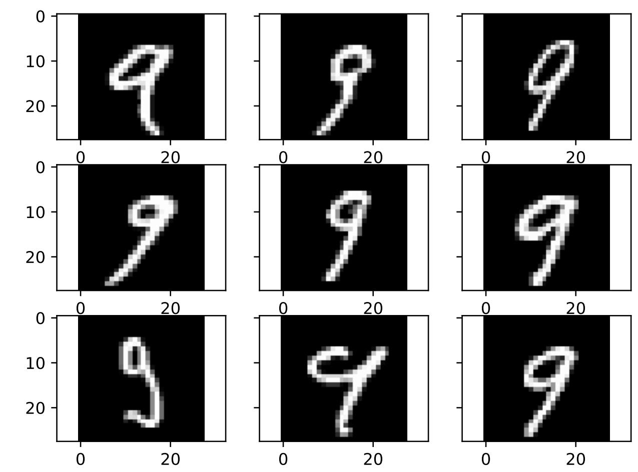
Claims:

- The bound is tight if $\alpha_x(y) = p_\theta(y \mid x)$,
- The sequence of likelihood values $L(\theta^{(t)}, \mathcal{T}^m)$, $t = 1, 2, \ldots$ is increasing, and the sequence $\alpha^{(t)}$, $t = 1, 2, \ldots$ is convergent (under mild assumptions).

3. The Expectation Maximisation Algorithm

Example: Latent mode model (mixture) for images of digits

- $x = \{x_i \mid i \in D\}$ image on the pixel domain $D \in \mathbb{Z}^2$,
- $x_i \in \mathcal{B} = \{0, 1, 2, \ldots, 255\}$
- $k \in K$ latent variable (mode indicator),
- joint distribution - Naive Bayes model



$$p(x, k) = p(k) \prod_{i \in D} p(x_i \mid k)$$

Learning problem: Given i.i.d. training data $\mathcal{T}^m = \{x^j \mid j = 1, 2, \ldots, m\}$, estimate the mode probabilities $p(k)$ and the conditional probabilities $p(x_i \mid k)$, $\forall x_i \in \mathcal{B}$, $k \in K$ and $i \in D$.

3. The Expectation Maximisation Algorithm

Applying the EM algorithm: Start with some model $p^{(0)}(k)$, $p^{(0)}(x_i \mid k)$ and iterate the following steps until convergence.

E-step Given the current model estimate $p^{(t)}(k)$, $p^{(t)}(x_i \mid k)$, compute the posterior mode probabilities for each image x in the training data $\mathcal{T}^m$

$$\alpha_x^{(t)}(k) = p^{(t)}(k \mid x) = \frac{p^{(t)}(k) \prod_{i \in D} p^{(t)}(x_i \mid k)}{\sum_{k'} p^{(t)}(k') \prod_{i \in D} p^{(t)}(x_i \mid k')}.$$

M-step Re-estimate the model by solving

$$\sum_{x \in \mathcal{T}^m} \sum_{k \in K} \alpha_x^{(t)}(k) \Big[\log p(k) + \sum_{i \in D} \log p(x_i \mid k) \Big] \to \max_p$$

This gives

$$p^{(t+1)}(k) = \frac{1}{m} \sum_{x \in \mathcal{T}^m} \alpha_x^{(t)}(k)$$

$$p^{(t+1)}(x_i = b \mid k) = \frac{\sum_{x \in \mathcal{T}^m :\, x_i = b} \alpha_x^{(t)}(k)}{\sum_{x \in \mathcal{T}^m} \alpha_x^{(t)}(k)}$$

3. The Expectation Maximisation Algorithm

Additional reading:

Schlesinger, Hlavac, Ten Lectures on Statistical and Structural Pattern Recognition, Chapter 6, Kluwer 2002 (also available in Czech)

Thomas P. Minka, Expectation-Maximization as lower bound maximization, 1998 (short tutorial, available on the internet)